



TITLE:

# Circle Generation on Two-Dimensional Cellular Automata (Algebraic Systems and Theoretical Computer Science)

AUTHOR(S):

Watanabe, Satoru; Okawa, Satoshi

---

CITATION:

Watanabe, Satoru ...[et al]. Circle Generation on Two-Dimensional Cellular Automata (Algebraic Systems and Theoretical Computer Science). 数理解析研究所講究録 2012, 1809: 161-168

ISSUE DATE:

2012-09

URL:

<http://hdl.handle.net/2433/194461>

RIGHT:

# Circle Generation on Two-Dimensional Cellular Automata

渡辺 識\*      大川 知†  
Satoru Watanabe      Satoshi Okawa

## 1 Introduction

In a dot matrix method, as a pattern or a figure are obtained by dots on lattices of fixed size, several dot patterns for each pattern should be prepared depending on the screen size. In recent years, as a method which is independent from a size of screen size, several methods to draw some patterns on a screen which is two-dimensional cellular automata have been proposed[4][6]. Komatsu has proposed a method to draw a square of maximum size in the center of the screen. Watanabe and Okawa have proposed another method to draw a square and a diamond of maximum size in the center of the screen[6]. In those methods, it is difficult to generate a curve or a roundish shape.

In this paper, we review the definitions of pattern and pattern generation, and we investigate to display a circle pattern of maximum size in the center of the screen which is two-dimensional cellular automata. First, we define two-dimensional patterns as equivalence classes which are obtained by the similarity relation defined by movings and scaling on two-dimensional plane. For an  $m \times n$  screen, we define a pattern generation to display with appropriate size (and position) in the screen. Next, we discretize the screen, and we define a pattern generation on the discretized screen. Furthermore, we study a correspondence between the discretized screen and cellular automata, and the pattern generation on the cellular automata. In the last part, we explain how to count square steps and we obtain a method to draw a circle on two dimensional cellular automata by using the method to count square steps.

## 2 Pattern Generation

For a set of real numbers  $\mathbb{R}$ , a two-dimensional plane is denoted by  $\mathbb{R} \times \mathbb{R}$ . A set  $F \subseteq \mathbb{R} \times \mathbb{R}$  is called a two-dimensional figure, and a set of all two-dimensional figures is denoted by  $\mathcal{F}$ , that is  $\mathcal{F} = \{F | F \subseteq \mathbb{R} \times \mathbb{R}\}$ . A figure which is obtained with moving  $F$  by  $d \in \mathbb{R} \times \mathbb{R}$  is denoted by  $F + d = \{p + d | p \in F\}$ , and a figure which is obtained with extending  $F$  by  $a$  ( $a > 0$ ) times is denoted by  $a \cdot F = \{a \cdot p | p \in F\}$ . We define mappings  $S_d$  and  $Z_a$  as follows, respectively.

$$S_d(F) = F + d, \quad Z_a(F) = a \cdot F.$$

We define a similarity relation  $\sim$  on  $\mathcal{F}$  using  $S_d$  and  $Z_a$  as follows.

For  $F_1, F_2 \in \mathcal{F}$ ,

$$F_1 \sim F_2 \Leftrightarrow F_2 = S_d Z_a(F_1) (= aF_1 + d).$$

---

\*The University of Aizu

†The University of Aizu

The relation  $\sim$  is an equivalence relation on two-dimensional figures. We define a pattern as a equivalence class using this relation as follows.

**Definition 1**

For a figure  $F$ , a pattern  $[F]$  containing  $F$  is defined by

$$[F] = \{F' | F' \sim F\},$$

and a set of patterns  $\mathcal{P}$  is defined by

$$\mathcal{P} = \mathcal{F} / \sim = \{P | P = [F], F \in \mathcal{F}\}.$$

For any  $m, n > 0$ ,  $[0, m] \times [0, n] \subseteq \mathbb{R} \times \mathbb{R}$  is called a screen of size  $m \times n$ , and it denoted by  $C_{m \times n}$ , where  $[a, b]$  is an interval  $\{x | a \leq x \leq b\}$ .

**Definition 2**

For a pattern  $P \in \mathcal{P}$  assuming  $P = [F]$ , generation of  $P$  on  $C_{m \times n}$  is to obtain a set  $D \subseteq C_{m \times n}$  which satisfies following conditions.

1.  $\exists a, d \quad D = S_d Z_a(F),$
2.  $\forall \epsilon, \epsilon' > 0 \quad S_{d+\epsilon} Z_{a+\epsilon'}(F) \not\subseteq C_{m \times n}.$

Following discussion, we assume that  $m$  and  $n$  are integers for simplicity. When we display a figure on a screen, the screen has to be discretized, so we discretize  $C_{m \times n}$  by dividing the width and the length by  $m - 1$  and  $n - 1$  respectively. In this process, for each lattice point  $p$ , a copy of small screen is set on it. The small screen at the leftmost and the bottom position of the discretized screen is  $c_{0,0}$ , and a screen which is positioned in the  $i$ th position from the left side of the array and  $j$ th position from the bottom of the array is described by  $c_{i,j}$ , that is  $c_{i,j} = C_{[i-0.5, i+0.5] \times [j-0.5, j+0.5]}$ .

We define the screen  $C_{m,n}$  which is obtained by discretizing  $C_{m \times n}$  as follows,

$$C_{m,n} = \{c_{i,j} | 0 \leq i \leq m, 0 \leq j \leq n, \quad i, j \in \mathbb{N}\}.$$

We define a pattern generation on the discretized screen as follows.

**Definition 3**

For a pattern  $P = [F] \in \mathcal{P}$ , generation of  $P$  on  $C_{m,n}$  is to obtain the following set  $D' \subseteq C_{m,n}$ ,

$$D' = \{c_{i,j} | c_{i,j} \cap D \neq \emptyset\}.$$

### 3 Implementation with Cellular Automata

Two-dimensional cellular automata consist of copies of a finite automaton (cell) each of which is positioned at lattice point  $(i, j)$ . Each cell changes its own state to the state which is determined by a function with its own state and the adjacent cells' states. We call the own and adjacent cells *neighbors*, the function to determine the next state according to neighbors' states is called a local map. Each cell is expressed by  $a_{i,j}$ , which means  $i$ th row and the  $j$ th column from the leftmost lowest cell. The interval of updating state is called a *step*. Formally, a two-dimensional cellular automaton  $\mathcal{M}$  is defined as follows,

$$\mathcal{M} = (M, Q, \sigma, N),$$

where  $M \subseteq \mathbb{Z} \times \mathbb{Z}$  is a set of coordinates where cells exist (we assume it is connected.  $\mathbb{Z}$

means a set of integers),  $Q$  is a set of states,  $\sigma : Q \times Q^{|N|-1} \rightarrow Q$  is a local map, and  $N$  is a set of neighbors. In this paper, we consider the automata which consist of  $m$  cells widthways and  $n$  cells lengthways and we call them  $m \times n$  cellular automata. We assume  $N$  as Neumann neighborhood, namely consisting of the own, upper, lower, right and left cells. In an initial configuration of  $\mathcal{M}$ ,  $a_{0,0}$  is in an active state, and all other cells are in a quiescent state.

By regarding each cell  $a_{i,j}$  as  $c_{i,j}$  in the discretized screen  $C_{m,n}$ , the set  $M$  can be regarded as the discretized screen  $C_{m,n}$ , and then, an  $m \times n$  cellular automaton can be denoted as follows,

$$\mathcal{M} = (C_{m,n}, Q, \sigma, N).$$

Therefore, we regard a problem to generate  $P$  on  $C_{m,n}$  as a problem to generate  $P$  on a cellular automaton  $\mathcal{M}$ , that is, a problem to construct  $\mathcal{M}$  which generates  $P$ .

To construct such  $\mathcal{M}$  is to provide  $\sigma$  which specifies  $D' \subseteq C_{m,n}$  at a certain time starting from the initial configuration. Here,  $D'$  is specified by letting  $a_{i,j}$  be in a special state  $s$  if  $a_{i,j} \in D'$ .

## 4 How to Count Square Steps

We explain propagation of signals among cells and how to count square steps in two-dimensional cellular array.

When a next cell of a cell in a specific state  $s$  changes its own state to  $s$  at  $k$  steps, we call the signal specified by  $s$  propagates at speed  $1/k$ . A cell can send signals upper, lower, right, and left directions.

To draw a circle pattern, we need to count  $i^2$  steps for each  $i$ . We will explain how to count square steps as follows[8]. Cell  $a_{0,0}$  sends Signal  $s$  with speed  $1/1$  to the right. When Signal  $s$  reaches a cell at which it has not reached yet, and then the cell sends back Signal  $\bar{s}$  to  $a_{0,0}$  with speed  $1/1$ . After receiving Signal  $\bar{s}$ , cell  $a_{0,0}$  sends Signal  $s$  to the right again. By repeating this, cell  $a_{i,0}$  receives Signal  $s$  just in  $i^2$  steps as shown in Figure 1. In the following discussion, we call this method *Square(s)*.

By following argument, it is clear that this method counts  $i^2$  steps for  $i > 0$ . Assume that we can count  $i^2$  steps by *Square(s)* by sending Signal  $s$  and Signal  $\bar{s}$  between  $a_{0,0}$  and  $a_{i,0}$ . After receiving Signal  $s$ ,  $a_{i,0}$  sends Signal  $\bar{s}$  to  $a_{0,0}$ . After receiving Signal  $\bar{s}$ ,  $a_{0,0}$  sends Signal  $s$  to the right direction again.  $a_{i,0}$  receives Signal  $s$ , and the next cell  $a_{i+1,0}$  receives Signal  $s$  at the next step.  $a_{i+1,0}$  receives Signal  $s$  with  $2i + 1$  steps after that  $a_{i,0}$  receives Signal  $s$  for the first time, that is,  $a_{i+1,0}$  can receive Signal  $s$  with  $i^2 + 2i + 1 = (i + 1)^2$  steps. Therefore, we can count square steps by using the method *Square(s)*.

## 5 Circle Pattern Generation on Cellular automata

We investigate a method to generate a circle pattern of maximum size in the center of a given  $m \times n$  cellular automaton. In the following example, we assume that  $m > n$ , and the area of the maximum square form in the center of the screen, cell  $O$ ,  $x$ -axis, and  $y$ -axis are obtained beforehand as shown in Figure 2. We generate an inscribed circle on the square.

### 5.1 Basic Concept of a Circle Pattern Generation

For each cell  $P$  in the square in Figure 3, we can find whether  $P$  is in the circle by checking whether  $\overline{PO} \leq \overline{KO}$ . This check can be done by counting  $\overline{PQ}^2 + \overline{QO}^2$  and  $\overline{KO}^2$ , which is explained as follows.

We can count  $\overline{PQ}^2$  steps by *Square(s<sub>1</sub>)* with Signal  $s_1$  from  $P$ . After receiving Signal  $s_1$ , cell  $Q$  send Signal  $s_2$  to cell  $O$ , and then we can count  $\overline{QO}^2$  steps by *Square(s<sub>2</sub>)* with Signal  $s_2$  from

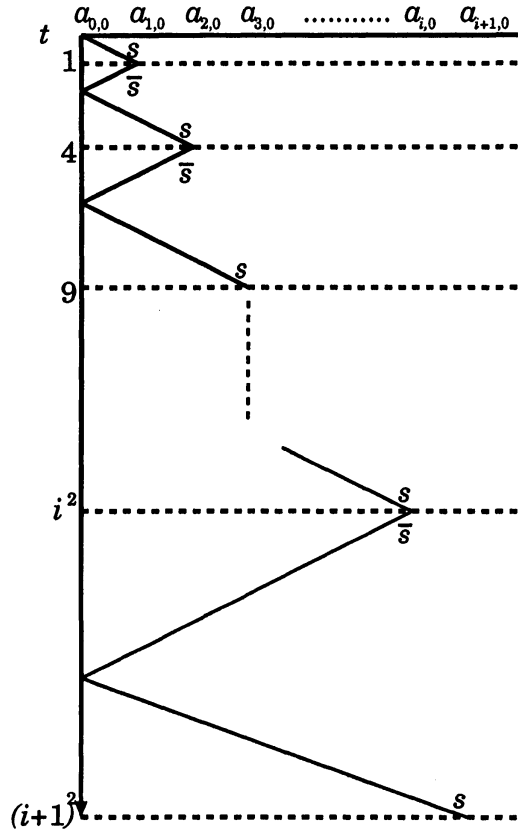


Figure 1: square signal

Q. Combining these counting, we can count  $\overline{PQ}^2 + \overline{QO}^2$  steps in total. Cell K sends Signal  $s_3$  to cell O, and then we can count  $\overline{KO}^2$  steps by  $Square(s_3)$  with Signal  $s_3$ .

By comparing these numbers of the steps, it is determined whether P is an internal cell of the circle or not. That is, if  $\overline{PQ}^2 + \overline{QO}^2 \leq \overline{KO}^2$ , P is internal, and P is external, otherwise.

## 5.2 Method to Generate a Circle

We explain each process of the generation of a circle in the center of the screen as follows.

### (1) synchronizing for checking

First, we need to synchronize starting time of countings  $\overline{PQ}^2 + \overline{QO}^2$  and  $\overline{KO}^2$  to check whether cell P ( $a_{i,j}$ ) is an internal one of the circle or not.

The cell P sends Signal  $j$  with speed 1/1 and Signal  $k$  with speed 1/2 to the cell K via cell M simultaneously, as shown in Figure 4.

After receiving Signal  $j$ , cell K sends back Signal  $\bar{j}$  to cell P via cell M. Signal  $\bar{j}$  and Signal  $k$  reach cell P and cell K simultaneously, as shown in Figure 5.

### (2) checking of the cells

After receiving Signal  $\bar{j}$ , cell P sends Signal  $l_1$  to cell Q, and then  $\overline{PQ}^2$  steps is counted by  $square(l_1)$  with Signal  $l_1$ . After receiving Signal  $l_1$ , cell Q sends Signal  $l_2$  to the center cell

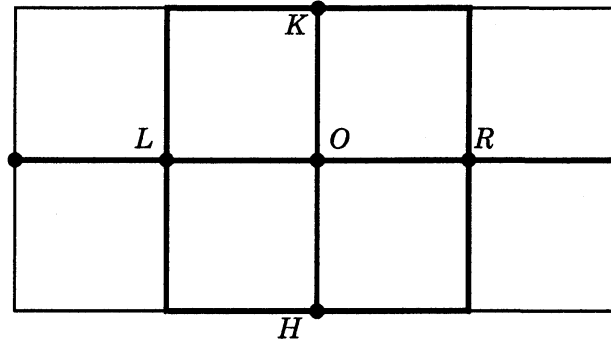


Figure 2: setting the center of the length and the width

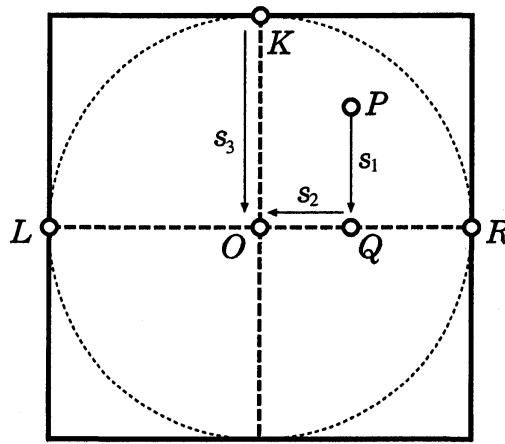


Figure 3: basic concept of a circle pattern generation

$O$ , and then we can count  $\overline{QO}^2$  steps by  $square(l_2)$  with Signal  $l_2$ , accordingly we can count  $\overline{PQ}^2 + \overline{QO}^2$  steps in total.

After receiving Signal  $k$ , cell  $K$  sends Signal  $m$  to cell  $O$ , and then we can count  $\overline{KO}^2$  steps by  $square(m)$  with Signal  $m$ , as shown in Figure 6.

Cell  $P$  is determined whether it is an internal point of the circle or not by checking which signal of  $l_2$  or  $m$  will arrive at the center cell  $O$  earlier. That Signal  $l_2$  arrives at the center cell  $O$  earlier, means  $\overline{PQ}^2 + \overline{QO}^2 \leq \overline{KO}^2$ . In this case, the cell  $P$  is internal of the circle, otherwise  $P$  is external of the circle.

If Signal  $l_2$  arrives at  $O$  earlier, the cell  $O$  sends signal  $y$  with speed  $1/1$  to the cell  $P$  via cell  $Q$ , otherwise the cell  $O$  send signal  $n$  with speed  $1/1$ , as shown in Figure 7. If cell  $P(a_{i,j})$  receives Signal  $y$ ,  $P$  changes its own state to the special state ' $s$ ' which indicates internal of the circle.  $P$  also sends Signal  $o$  to the lower direction to change states of lower cells of  $P$ . The cells which are passed by Signal  $o$  change their own state to the special state ' $s$ ', and then Signal  $o$  stops at  $Q$ . Furthermore,  $P$  sends a signal  $p$  to the next cell  $a_{i+1,j}$ , as shown in Figure 8. If cell  $P$  receives Signal  $n$ ,  $P$  sends Signal  $p$  to the lower cell  $a_{i,j-1}$ . In the cell  $a_{i+1,j}$  or  $a_{i,j-1}$ , same checking is performed repeatedly.

When Signal  $p$  arrives at the cell  $R$ , the checking finishes.

### (3) copies of state ' $s$ '

To generate whole circle form, copies of state ' $s$ ' are performed in the other quadrants. Cell

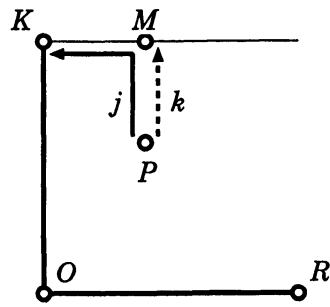


Figure 4: synchronizing1

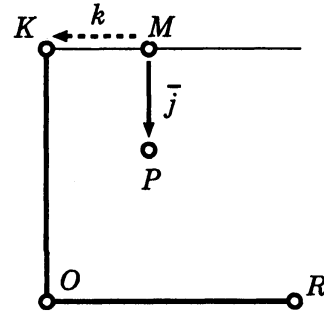


Figure 5: synchronizing2

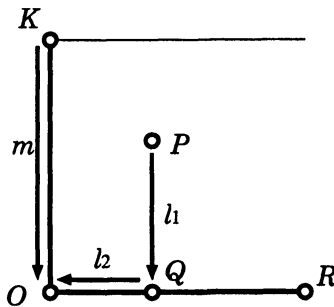


Figure 6: checking of the cells 1

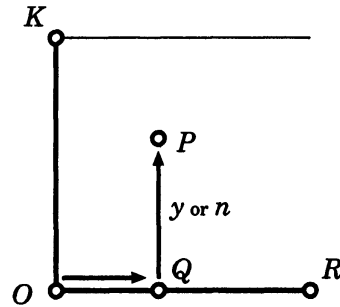


Figure 7: checking of the cells 2

$P$  sends Signal  $q$  and  $r$  with speed  $1/1$  with Signal  $l_1$  simultaneously, as shown in Figure 9.

After receiving Signal  $q$ , Cell  $Q$  sends Signal  $t$  with speed  $1/1$  to the upper and Signal  $u$  with speed  $1/3$  to the lower. After receiving Signal  $t$ , Cell  $P$  sends Signal  $\bar{t}$  with speed  $1/1$ . Signal  $\bar{t}$  and Signal  $u$  hit in cell  $P'$  which is in line symmetric position of cell  $P$ , as shown in figure 10. Similarly, cell  $P''$  is obtained in the second quadrant.

If cell  $P$  is internal of the circle, cell  $P'$  receives Signal  $y$  from cell  $O$  by the same way as  $P$ , and changes its own state to ' $s$ '.  $P'$  also sends Signal  $o'$  to the upper direction to change states of upper cells of  $P'$ . The cells which are passed by Signal  $o'$  change their own state to the special state ' $s$ ', and then Signal  $o'$  stops at  $Q$ . If cell  $P$  is external of the circle, cell  $P'$  receives Signal  $n$  from cell  $O$ , and then  $P'$  does nothing.

Similarly, the all internal cells are obtained in the second quadrant. In the third quadrant, the copies are obtained as follows. When the states of cell  $P'$  and  $P''$  become ' $s$ ', cell  $P'$  sends Signal  $v$  to the left and cell  $P''$  sends Signal  $w$  to the lower. A signal hits the another signal's trace, and then the hit point becomes cell  $P'''$ , as shown in Figure 11. The cells in the internal of the circle change their own state to the special state ' $s$ ' by the same way as the other quadrant. When the all states of the internal of the circle are copied into the all quadrant, we can obtain the circle.

## 6 Conclusion

In this paper, we reviewed the definition of a pattern and a pattern generation. Next, we discretized the screen, and defined a pattern generation on the discretized screen. Second, we studied a correspondence between the discretized screen and cellular automata, and we studied the pattern generation on the cellular automata. Furthermore, we explained a method to count square steps by sending signals. In the last part, we obtained a method to draw a circle on two-dimensional cellular automata, by using the method to count square steps.

In this paper, we concentrated only the generation of a circle, and did not consider com-

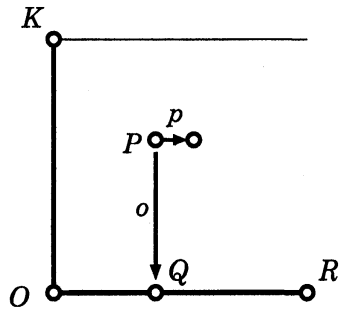


Figure 8: changing states of the internal cells

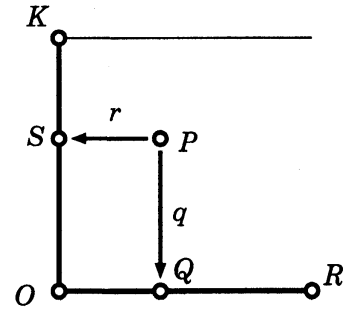


Figure 9: setting symmetric position1

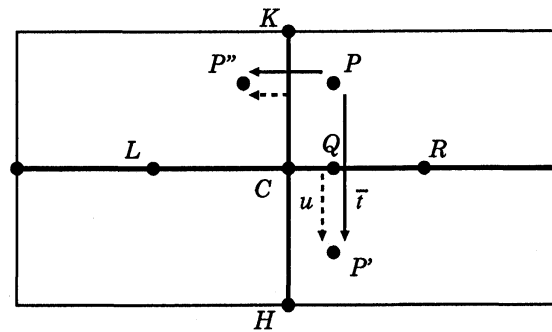


Figure 10: setting symmetric position2

plexity issue. In our construction of cellular automaton, by checking only cells near to the circumference of the circle, we have cellular automaton working in  $O(r^3)$  time. ( $r$  means a radius of a circle) Furthermore, by drawing a circle and copying the states which indicates the internal of the circle simultaneously, the circle pattern is obtained more quickly.

At present, we have to consider a method for pattern generation according to an individual figure. If any pattern can be generated using a standardized method, we can use it in a display such as an electric bulletin board or a printer.

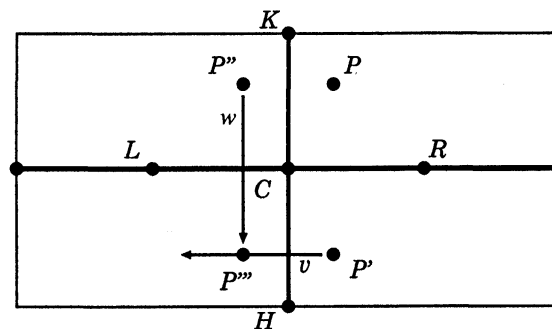


Figure 11: setting symmetric position3



# reference

- [1] Y.Mizuno, A New Solution of the Firing Squad Synchronization Problem for Two Dimensional Rectangle Arrays, *University of Aizu, Grad.Thesis.*, March, 2005.
- [2] M.Teraoka, et al. A Design of Generalized Optimum-Time Firing Squad Synchronization Algorithm for Two-Dimensional Cellular Arrays, *The 18th Annual Conf.of Japanese Society for Artificial Intelligence*, 3H1-04,2004.
- [3] A.Settle and J.Simon, Smaller solutions for the firing squad, *Theoretical Computer Science*, 276, pp.83-109, 2002.
- [4] T.Komatsu, Pattern Generation in Two Dimension Plane, *University of Aizu, Grad.Thesis.*, March, 2008.
- [5] S.Wolfram, Two-Dimensional Cellular Automata, *Journal of Statistical Physics*, vol 38, p901-946, March 1985.
- [6] S.Watanabe and S.Okawa, Pattern Generation on Two Dimensional Cellular Automata, *Forum on Information Technology 2009*, A-023.
- [7] J. Mazoyer, A six state minimal time solution to the firing squad synchronization problem, *Theoretical Computer Science*, vol.50, pp.183-238,1987
- [8] Y. Nishitani, Firing Synchronization Problems for Patterns specified by the Sub-General Position, *Cellular Automaton Seminar in University of Aizu* ,2010